Iterative Learning Control of Minimum Energy Path Following Tasks for Second-Order MIMO Systems: An Indirect Reference Update Framework

Yiyang Chen, *Member, IEEE,* Yiming Wang and Christopher T. Freeman

***Abstract*—In a large range of manufacturing tasks, the design objective is characterised as following a given path defined in space. In these applications, the tracking time of any particular position along the path is not specified, so an appropriate motion profile can be chosen among its admissible solutions to improve its tracking performance. This paper develops an indirect reference update framework that maximizes accuracy while embedding practical constraints. An optimal path planning problem, incorporating system constraints, is formulated and can be solved using a discretized approach to derive a motion profile that minimizes control energy for a broad spectrum of industrial tasks. To satisfy robustness concerns, an iterative learning control (ILC) algorithm with an indirect reference update framework is designed to improve the accuracy and robustness of path following. It is evaluated on a gantry robot test platform, and the results illustrate superior levels of practical performance in terms of energy reduction and path following accuracy compared with existing approaches.**

***Index Terms*—iterative learning control, path planning, constraint handling, optimization.**

I. Introduction

ROBOTS are extensively applied in modern manufacturing due to their efficiency and precision [1]. A significant class of industrial operations are characterized by directing the robot's end-effector along a specified trajectory, such as manufacturing, assembly [2], welding [3], drilling [4] and electro-hydraulic rolling [5]. These tasks focus on precisely following a path, rather than achieving a position tracking objective at each individual time instant as in the case of trajectory tracking [6]. Therefore, the end-effector's moving speed along the path is not specified by a unique motion profile [7]. In this path following problem setup, the temporal tracking requirement component is removed and significant design freedom is released compared to trajectory tracking with a fixed motion profile [8]. However, the flexibility of choosing a time-varying motion profile from the large set of admissible choices also brings design challenges.

Due to the elimination of the temporal tracking requirement, there exist an infinite number of motion profiles for a given path following task. Existing research has attempted to capitalise on the flexibility of choosing an appropriate motion profile to achieve performance benefits. The work in [9]–[13] leveraged the flexibility in path planning to enhance task execution, e.g., minimum traveling time, via robotic motion planning techniques. Parallel work in [14]–[16] considered the collision avoidance problem in path planning using dynamic or heuristic approaches, which may change the reference profiles. However, all the aforementioned work used nominal system models to compute the input signal and motion profile solutions, and their practical path following accuracy was therefore sensitive to the level of model uncertainty. Moreover, they primarily focused on tracking, but did not address economic issues such as energy reduction. However, these issues are key performance indicators in modern industry [17], [18].

In terms of robustness, various control algorithms have been designed to improve path following performance in practice. In [19]–[21], the class of vehicle steering problems were studied with respect to both model uncertainty and random disturbance [22], [23]. A real-time motion profile was obtained via the feedback of measured data, e.g., displacement and traveling direction, such that the vehicle asymptotically follows its given path. Aiming at improving the path following accuracy, iterative learning control (ILC) was introduced in order to update the input signal using the measured data from past trials, such that the path following error gradually reduces to a sufficiently small value [24]–[26]. Related work in [27], [28] applied ILC to a class of systems with constraints for path following tasks. Research in [29]–[31] applied energy-efficient or ILC-based algorithms to path planning tasks. However, like most ILC approaches for spatial tracking, only linear dynamics were considered.

The stabilization of nonlinear systems while improving system performance has long been a challenging problem. Related research [32]–[34] has focused on integrating nonlinear control techniques and data-driven algorithms to improve system performance while ensuring system stability. In terms of data-driven methods, various intelligent control algorithms combined with data-driven techniques have been proposed to control complex systems. In [35], [36], Neural Network (NN) based control strategies, such as Q-learning and adaptive implicit inverse control, were applied in the control and optimization of time-delay and nonlinear systems. However, the stability analysis of data-driven Model Free Control (MFC) algorithms is a key issue and is closely related to the mathematical model of the controlled process. A discussion of this

This research paper was invested by National Natural Science Foundation of China under Grant 62103293 and Natural Science Foundation of Jiangsu Province under Grant BK20210709. (Corresponding author: Yiyang Chen)

Yiyang Chen and Yiming Wang are with the School of Mechanical and Electrical Engineering, Soochow University, No. 8, Jixue Road, Suzhou, Jiangsu 215137, China. (e-mail: yychen90@suda.edu.cn; ymwang0105@stu.suda.edu.cn)

Christopher T. Freeman is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom. (e-mail: ctf1@soton.ac.uk)

topic can be found in [37].

To expand the class of admissible robotic applications, this paper considers a minimum energy path following problem with system constraints for industrial robots, e.g., robotic manipulator [38], [39] and Cartesian coordinate robots [40], [41], which repetitively perform the same task. The technique of ILC has been applied to this application area in several previous studies, where it has proved capable of leveraging learning to outperform other control methods in terms of path following accuracy [42], [43], implementation simplicity and computational load reduction [44], [45]. To address the minimum energy problem, the work in [46], [47] combined optimal path planning with ILC to yield a comprehensive algorithm with an indirect reference update framework, which provides path following accuracy, energy reduction and constraint handling. In [48], a low-cost fuzzy control scheme was proposed that employed ILC algorithms and a unified design method focused on Takagi-Sugeno proportional-integral fuzzy controllers. In [49], [50], the problem of 2D path following minimum time was addressed using ILC. However, the inner structure of the optimization problem varies for each trial due to the nominal model update, which hence increased the computation load. Previous ILC approaches have imposed restrictions on the form of path that can be followed which lead to a suboptimal solution. For example, the leading approach [40] stipulated additional 'point-to-point' timings throughout the motion that complicated the design and implementation.

In order to liberate available degrees of freedom in control time duration, enhance the capability of optimizing performance indicators, and simplify the conventional design framework of Optimal ILC (OILC), this paper introduces an indirect reference update framework employing ILC to address the minimum energy path following problem. This gives rise to a highly coupled system input and reference profile which motivates a reference profile update ILC law instead of the input signal update law used in conventional ILC. The reference profile is then linked to the input signal using the global solution of minimum energy path following problems. In this way, this paper obtains an indirect reference update framework of ILC embedding both a reference profile update and minimum energy path following solution. This framework not only provides a minimum energy path strategy, but also improves the path following accuracy and robustness in the presence of model uncertainty. The paper's contributions include:

1) The path following task is defined for a general class of robotic system, embedding system constraints with coordinates in a spatial domain, and a minimum energy path planning problem is formulated.
2) The path planning component is redefined as a convex optimization problem within the spatial domain, yielding a global solution through a discretized methodology.
3) A comprehensive ILC algorithm incorporating an indirect reference update framework is developed to enhance the precision of path tracking in the presence of model uncertainties.
4) The proposed algorithm is verified on a gantry robot to demonstrate feasibility, and a comparison with unplanned motion profiles is also made in terms of energy reduction.

The following notation is used: $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the sets of n dimensional real vectors and $n \times m$ real matrices respectively; $L_2^\ell[a,\ b]$ denotes the space of $\mathbb{R}^\ell$ valued Lebesgue square-summable sequences defined on an interval $[a, b]$.

II. Problem Formulation

This section presents the system dynamics, and defines the path following task in a spatial domain with system constraints. The minimum energy path following problem is then formulated for the class of industrial robots.

A. System Dynamics

An ℓ-input, m-output time-invariant robotic system with second order dynamics is given by

$$\begin{aligned} R(q(t))u(t) &= M(q(t))\ddot{q}(t) + C(q(t),\dot{q}(t))\dot{q}(t) + D(q(t)),\\ y(t) &= q(t), \end{aligned} \tag{1}$$

where $t \in [0,\ T]$ is the time index with $0 < T < \infty$ denoting the trial length; $R \in \mathbb{R}^{m\times\ell}$ is the control matrix, $M \in \mathbb{R}^{m\times m}$ is the mass matrix, $C \in \mathbb{R}^{m\times m}$ is the centrifugal matrix which is linear in $\dot{q}$, and $D \in \mathbb{R}^m$ is the force dependent on configuration; $u(t) \in \mathbb{R}^\ell$ and $y(t) \in \mathbb{R}^m$ are the input and output respectively; $q(t) \in \mathbb{R}^m$ denotes the state of the system. This system represents a general class of robotic applications widely used in manufacturing, and its input output relationship can be represented by

$$y = G(u), \tag{2}$$

where the operator G maps the input signal $u \in L_2^\ell[0,\ T]$ to the output signal $y \in L_2^m[0,\ T]$. The input and output Hilbert spaces $L_2^\ell[0,\ T]$ and $L_2^m[0,\ T]$ are defined with inner products and associated induced norms

$$\langle u,\ v\rangle = \int_0^T u^\top(t)v(t)\mathrm{d}t,\ \ \|u\| = \sqrt{\langle u,\ u\rangle}, \tag{3}$$

$$\langle x,\ y\rangle = \int_0^T x^\top(t)y(t)\mathrm{d}t,\ \ \|y\| = \sqrt{\langle y,\ y\rangle}. \tag{4}$$

B. Path Following Task

A path is described as a subset of points in the output space, $\mathbb{R}^m$, defined by a continuous function, $\tilde{r}$, which maps each spatial position s in $[0,\ 1]$ to a point $\tilde{r}(s)$ in the output space, i.e. $\tilde{r} : s \in [0,\ 1] \to \tilde{r}(s) \in \mathbb{R}^m$. In a path following task, the reference profile $\tilde{r}$ is given as *a priori* and hence is independent of temporal information. In addition, the system input and output signals can be alternatively defined using spatial coordinates as

$$\tilde{u} \in L_2^\ell[0,\ 1],\ \tilde{y} \in L_2^m[0,\ 1]. \tag{5}$$

In general, the definition of a path following task is given as

Definition 1. **Path Following Task:** the end-effector of a robot first starts from an initial position $\tilde{r}(0)$, and must accurately follow a path defined by $\tilde{r}$ and reach a terminal position $\tilde{r}(1)$ within a time period T.

According to Definition 1, the end-effector initiates its motion from the position $\tilde{r}(0)$ at the initial time instant $t = 0$, with the initial state condition that $q(0) = 0$, and proceeds to reach the position $\tilde{r}(1)$ at the terminal time instant $\hat{T}$, with the constraint that $0 < \hat{T} \leqslant T$. Upon reaching the terminal position, the end-effector ceases movement, thus for the interval $[\hat{T}, T]$, the following conditions hold:

$$q(t) = \hat{q},\ u(t) = \hat{u},\ \dot{y}(t) = 0,\ t \in [\hat{T},\ T], \tag{6}$$

where $\hat{q}$ represents the constant value at rest, and $\hat{u}$ is the solution to the linear equation $R(\hat{q})u(t) = D(\hat{q})$ pertaining to $u(t)$.

To connect the spatial and temporal variables, a motion profile θ is specified to determine the speed of the end-effector along the path. The motion profile is defined as a bijective continuous function, mapping each time instant t in $[0,\ \hat{T}]$ to a unique spatial variable s in $[0,\ 1]$ i.e. $\theta : t \in [0,\ \hat{T}] \to s \in [0,\ 1]$ as follows:

$$s = \theta(t),\ \dot{\theta}(t) \geqslant 0,\ t \in [0,\ \hat{T}] \tag{7}$$

with initial and terminal conditions

$$\theta(0) = 0,\ \theta(\hat{T}) = 1,\ \dot{\theta}(0) = 0,\ \dot{\theta}(\hat{T}) = 0. \tag{8}$$

Using the motion profile, any given signal can be transformed between temporal and spatial coordinates as

$$\begin{aligned} &r(t) = \tilde{r}(\theta(t)),\ t \in [0,\ \hat{T}], \\ &\tilde{r}(s) = r(\theta^{-1}(s)),\ s \in [0,\ 1]. \end{aligned} \tag{9}$$

Using this spatial transformation, the design objective of the path following task is to design an input signal u to guarantee the output trajectory $y = G(u)$ accurately following the given path defined by $\tilde{r}$, i.e.,

$$y = G(u),\ y = r,\ r(t) = \tilde{r}(\theta(t)),\ t \in [0,\ \hat{T}],\ \theta \in \Theta, \tag{10}$$

where the motion profile θ belongs to the admissible set

$$\begin{aligned} \Theta = \{\theta : [0,\ \hat{T}] \to [0,\ 1] :\ &\theta(0) = 0,\ \theta(\hat{T}) = 1, \\ &\dot{\theta}(0) = 0,\ \dot{\theta}(\hat{T}) = 0,\ \dot{\theta}(t) \geqslant 0,\ t \in [0,\ \hat{T}]\} \end{aligned} \tag{11}$$

for all possible motion profiles.

C. System Constraints

In industrial tasks, there are various system constraints that restrict system performance. For instance, an input constraint represents the physical limit of the system's tolerance, beyond which hardware damage may occur. Therefore, it is imperative to keep these variables within their corresponding constraints while performing these industrial tasks.

In the path following task, the primary system constraints associate with the input signal u, speed $\dot{\theta}$ and acceleration $\ddot{\theta}$. Hence, the following two spatial coordinate transformations are introduced to represent the speed and acceleration,

$$a(s) = \ddot{s}(\theta^{-1}(s)) = \ddot{s}(t),\ b(s) = \dot{s}^2(\theta^{-1}(s)) = \dot{s}^2(t) \tag{12}$$

for $s \in [0,\ 1]$. The path following task system constraints have the form

$$(\tilde{u},\ a,\ b) \in \mathcal{C}, \tag{13}$$

where $\mathcal{C}$ is a convex set consisting of all admissible values of the triplet $(\tilde{u},\ a,\ b)$.

Remark 1. The system constraints in (13) should be chosen based on practical concerns, such as the performance limit of the robot and the working environment. See [51] for some exemplary system constraints required by practical applications. In particular, input saturation constraints are taken into consideration in the experimental section as they are required to prevent damage to the gantry robot.

D. Minimum Energy Path Following Problem

The design objective (10) naturally allows an infinite number of alternative solutions for the path following task, which provides significant design freedom. This sort of freedom enables a suitable choice of the input u and motion profile θ, to optimize some performance index and maintain the objective (10) at the same time. Therefore, a path following problem can be formulated as

$$\begin{aligned} \min_{\theta}\quad & f(u, y) \\ \text{s.t.}\quad & y = G(u),\ \tilde{y} = \tilde{r}, \\ & y(t) = \tilde{y}(\theta(t)),\ t \in [0,\ \hat{T}], \\ & u(t) = \hat{u},\ t \in [\hat{T},\ T], \\ & 0 < \hat{T} \leqslant T,\ \theta \in \Theta,\ (\tilde{u}, a, b) \in \mathcal{C}. \end{aligned} \tag{14}$$

In optimization problem (14), the decision variable is θ which represents the system performance can be optimized by the motion profile. The cost function $f(u, y)$ represents the target performance index, which might vary according to different tasks. As an example, to minimize the peak input, $f(u, y)$ is chosen as

$$f(u, y) = \|u\|_\infty,$$

and to minimize the output acceleration, $f(u, y)$ is chosen as

$$f(u, y) = \|\ddot{y}\|.$$

In this paper, the path following task is considered, and the target performance index is selected as the overall control energy of this task, i.e.,

$$f(u, y) = \|u\|^2.$$

This choice has significant economic effect in manufacturing. In industry, the number of robotic tasks completed every day is large, so even a small degree of energy reduction in a single task accumulates to a large energy saving. Hence, a minimum energy path following problem is formulated as

$$\min_{\theta} \quad \|u\|^2 \tag{15a}$$

$$\text{s.t.} \quad y = G(u), \tag{15b}$$

$$y(t) = \tilde{y}(\theta(t)) = \tilde{r},\ t \in [0,\ \hat{T}] \tag{15c}$$

$$u(t) = \hat{u},\ t \in [\hat{T},\ T], \tag{15d}$$

$$0 < \hat{T} \leqslant T, \tag{15e}$$

$$\theta \in \Theta, \tag{15f}$$

$$(\tilde{u}, a, b) \in \mathcal{C}. \tag{15g}$$

In the minimum energy path following problem (15), the constraints have the following interpretation:

1) Eq. (15b) embeds the system dynamics.
2) Eq. (15c) converts the trajectory tracking task from the temporal domain to the spatial domain during the movement interval.
3) Eq. (15d) converts the inputs from the temporal domain to the spatial domain after the movement has ceased.
4) Eq. (15e) specifies that the movement time interval does not exceed the feasible time interval.
5) Eq. (15f) specifies the feasible set of decision variable θ.
6) Eq. (15g) ensures that the admissible values of the triplet $(\tilde{u}, a, b)$ are in a convex set.

III. Path planning strategy

In general, the minimum energy path following problem (15) is non-linear and non-convex, which makes it difficult to solve. Consequently, this problem is reformulated into an equivalent convex optimization problem in the spatial domain, and a discretized approach is applied to solve the reformulated problem.

A. Problem Reformulation

Using the definition of motion profile (7), the condition (15c) gives rise to

$$y(t) = \tilde{y}(\theta(t)) = \tilde{r}(\theta(t)),\ t \in [0,\ \hat{T}],\ \theta \in \Theta, \tag{16}$$

and the derivatives of $y(t)$ is represented as

$$\begin{aligned} \dot{y}(t) &= \tilde{r}^{'}(s)\dot{\theta}(t), \\ \ddot{y}(t) &= \tilde{r}^{'}(s)\ddot{\theta}(t) + \tilde{r}^{''}(s)\dot{\theta}^2(t), \end{aligned} \tag{17}$$

where

$$\tilde{r}^{'} = \frac{\mathrm{d}\tilde{r}}{\mathrm{d}s},\ \tilde{r}^{''} = \frac{\mathrm{d}^2\tilde{r}}{\mathrm{d}s^2}. \tag{18}$$

Here the symbol $\dot{}$ represents the derivatives with respect to t, and the symbol $'$ represents the derivatives with respect to s. After the reformulation in the spatial domain, the initial state condition is converted to $\tilde{q}(0) = q(\theta^{-1}(0)) = 0$.

Using the above representations, the original problem (15) is equivalently reformulated from time domain to spatial domain via the change of variable scheme proposed in [52]. The derivatives (17) are substituted into the system dynamics (1) to yield the spatial representation of the system dynamics as

$$\tilde{R}(s)\tilde{u}(s) = \tilde{M}(s)\ddot{s} + \tilde{C}(s)\dot{s}^2 + \tilde{D}(s), \tag{19}$$

where $\tilde{R}(s)$, $\tilde{M}(s)$, $\tilde{C}(s)$ and $\tilde{D}(s)$ are constants in spatial domain for given reference profile $\tilde{r}$ denoted as

$$\begin{aligned} \tilde{R}(s) &= R(\tilde{r}(s)), \\ \tilde{M}(s) &= M(\tilde{r}(s))\tilde{r}^{'}(s), \\ \tilde{C}(s) &= M(\tilde{r}(s))\tilde{r}^{''}(s) + C(\tilde{r}(s))\tilde{r}^{'2}(s), \\ \tilde{D}(s) &= D(\tilde{r}(s)). \end{aligned}$$

The next proposition illustrates the equivalent representation of problem (15) in the spatial domain.

Proposition 1. The minimum energy path following problem (15) is equivalently reformulated in the spatial domain as the convex optimization problem

$$\min_{\tilde{u},a,b} \int_0^1 \frac{\tilde{u}^\top(s)\tilde{u}(s)}{\sqrt{b(s)}}\mathrm{d}s + \hat{E}(\hat{T}, \hat{u}) \tag{20a}$$

$$\text{s.t. } \tilde{R}(s)\tilde{u}(s) = \tilde{M}(s)a(s) + \tilde{C}(s)b(s) + \tilde{D}(s), \tag{20b}$$

$$b^{'}(s) = 2a(s), \tag{20c}$$

$$b(s) \geqslant 0,\ s \in [0,\ 1], \tag{20d}$$

$$\int_0^1 \frac{1}{\sqrt{b(s)}}\mathrm{d}s \leqslant T, \tag{20e}$$

$$(\tilde{u}, a, b) \in \mathcal{C}. \tag{20f}$$

Proof. See Appendix A. □

By re-specifying the system model $G(\cdot)$ via the temporal-spatial coordination conversion mechanism as a linear constraint, Proposition 1 reformulates the problem (15) into a convex optimization problem (20). The decision variables become the spatial input $\tilde{u}$, spatial speed a and jerk b. In addition, any local optimal solution of this reformulated problem is guaranteed to be a global optimum. This appealing property enables a reliable computational path planning strategy to address the original design problem. The subsequent Proposition 2 delineates a method for computing the energy consumption term $\hat{E}(\hat{T}, \hat{u})$, ensuring its minimization.

Proposition 2. Suppose the control input $u(t) = \hat{u}$ during the interval $[\hat{T},\ T]$ being the least-squares solution to the linear equation $R(\hat{q})u(t) = D(\hat{q})$ pertaining to $u(t)$, is given by

$$\hat{u} = R^\top(\hat{q})(R(\hat{q})R^\top(\hat{q}))^{-1}D(\hat{q}), \tag{21}$$

then the energy consumption term $\hat{E}(\hat{T}, \hat{u})$ is minimum for any given $\hat{T}$.

Proof. See Appendix B. □

Proposition 2 effectively demonstrates that the optimal control input $\hat{u}$, which minimizes energy consumption during the interval $[\hat{T},\ T]$, corresponds to the least-squares solution of the equation $R(\hat{q})u(t) = D(\hat{q})$. Hence, with the optimization of $\hat{E}(\hat{T}, \hat{u})$ in objective (20a) is addressed elegantly, the complexity of the original optimization problem is significantly reduced.

B. Solution via Discretized Approach

The problem (20) is an infinite dimensional convex optimization problem, which is hard to solve directly. Therefore, it is converted into a finite dimensional problem using a discretized approach proposed in [52]. The variables $\tilde{u}(s)$, $a(s)$ and $b(s)$ are discretized at $(N+1)$ sample points, i.e., $s_i,\ i = 0, \ldots, N$, in $[0,\ 1]$ with equal distance between the adjacent pairs s_i and s_{i+1}, i.e.,

$$\Delta s = s_{i+1} - s_i,\ i = 0, \ldots, N-1. \tag{22}$$

Denote b_i as the value of $b(s)$ at s_i, i.e.,

$$b_i = b(s_i),\ i = 0, \ldots, N, \tag{23}$$

and a_i and u_i as the values of $a(s)$ and $\tilde{u}(s)$ at the middle of each sub-interval $[s_i,\ s_{i+1}]$, i.e.,

$$a_i = a(\frac{s_{i-1}+s_i}{2}),\ u_i = \tilde{u}(\frac{s_{i-1}+s_i}{2}),\ i = 1,\dots,N. \quad (24)$$

If a sufficiently high sampling frequency is chosen, then the discretized points a_i, s_i match their continuous counterparts over sub-intervals containing s_i. This leads to the assumption:

Assumption 1. The values of $a(s)$ and $\tilde{u}(s)$ are constant on each sub-interval $[s_i,\ s_{i+1}]$, i.e.,

$$a(s) = a_i,\ \tilde{u}(s) = u_i,\ s \in [s_i,\ s_{i+1}],\ i = 1,\dots,N. \quad (25)$$

Based on Assumption 1, the next proposition shows the discretized form of problem (20).

Proposition 3. The convex optimization problem (20) has the following discretized form

$$\min_{\tilde{u},a,b} \sum_{i=1}^{N} \frac{2\Delta s \cdot u_i^\top u_i}{\sqrt{b_{i-1}}+\sqrt{b_i}} + \hat{E}(\hat{T},\hat{u}) \quad (26a)$$

$$\text{s.t. } \tilde{R}_i u_i = \tilde{M}_i a_i + \tilde{C}_i \cdot \frac{b_{i-1}+b_i}{2} + \tilde{D}_i, \quad (26b)$$

$$b_i - b_{i-1} = 2a_i \cdot \Delta s,\ i = 1,\dots,N, \quad (26c)$$

$$b_i \geqslant 0,\ i = 0,\dots,N, \quad (26d)$$

$$\sum_{i=1}^{N} \frac{2\Delta s}{\sqrt{b_{i-1}}+\sqrt{b_i}} \leqslant T, \quad (26e)$$

$$(\tilde{u},a,b) \in \tilde{\mathcal{C}}, \quad (26f)$$

where the system parameters $\tilde{R}_i$, $\tilde{M}_i$, $\tilde{C}_i$ and $\tilde{D}_i$ are calculated as

$$\tilde{R}_i = \tilde{R}(\frac{s_{i-1}+s_i}{2}),\ \tilde{M}_i = \tilde{M}(\frac{s_{i-1}+s_i}{2}),$$
$$\tilde{C}_i = \tilde{C}(\frac{s_{i-1}+s_i}{2}),\ \tilde{D}_i = \tilde{D}(\frac{s_{i-1}+s_i}{2}). \quad (27)$$

Proof. See Appendix C. □

Problem (26) is the discretized form of the spatial optimization problem (20) and can be solved using numerical methods. The latter problem can be omitted if the problem is originally stated in discrete-time, and the former taken as the starting point. Problem (26) can be written in a standard optimization setup, which is shown in the next corollary.

Corollary 1. The discretized form (26) is rewritten as a second order cone programming (SOCP) problem as

$$\min_{\tilde{u},a,b,c,d,\hat{d}} \sum_{i=1}^{N} 2\Delta s \cdot d_i + \hat{E}(\hat{T},\hat{u}) \quad (28a)$$

$$\text{s.t. } \tilde{R}_i u_i = \tilde{M}_i a_i + \tilde{C}_i \cdot \frac{b_{i-1}+b_i}{2} + \tilde{D}_i, \quad (28b)$$

$$\left\| \begin{matrix} 2u_i \\ c_{i-1}+c_i-d_i \end{matrix} \right\|_2 \leqslant c_{i-1}+c_i+d_i, \quad (28c)$$

$$\left\| \begin{matrix} 2 \\ c_{i-1}+c_i-\hat{d}_i \end{matrix} \right\|_2 \leqslant c_{i-1}+c_i+\hat{d}_i, \quad (28d)$$

$$b_i - b_{i-1} = 2a_i \cdot \Delta s,\ i = 1,\dots,N, \quad (28e)$$

$$\left\| \begin{matrix} 2c_i \\ b_i-1 \end{matrix} \right\|_2 \leqslant b_i+1, \quad (28f)$$

$$b_i \geqslant 0,\ i = 0,\dots,N, \quad (28g)$$

$$\sum_{i=1}^{N} 2\Delta s \cdot \hat{d}_i = T, \quad (28h)$$

$$(\tilde{u},a,b) \in \tilde{\mathcal{C}} \quad (28i)$$

by introducing three more variables c, d, and $\hat{d}$.

Proof. See Appendix D. □

The problem (26) is reformulated into the SOCP problem (28), which combines features of linear programming (LP) and quadratic programming (QP), has strong versatility, and has a wide range of applicable scenarios. In addition, it can be readily solved using standard numerical solvers and parsers available in **MATLAB**, such as **SeDuMi** [53] and **YALMIP** [54]. Once the problem (26) is solved, the optimal input solution can be directly obtained, and the optimal motion profile can be reconstructed from the solution of a and b.

Remark 2. The path information is required while calculating the values of $\tilde{R}_i$, $\tilde{M}_i$, $\tilde{C}_i$ and $\tilde{D}_i$. The analytic approach is used to calculate the path reference derivatives, or alternatively they can be estimated using

$$\tilde{r}(\frac{s_{i-1}+s_i}{2}) = \frac{\tilde{r}(s_{i-1})+\tilde{r}(s_i)}{2},$$
$$\tilde{r}'(\frac{s_{i-1}+s_i}{2}) = \frac{\tilde{r}(s_i)-\tilde{r}(s_{i-1})}{\Delta s},$$
$$\tilde{r}''(\frac{s_{i-1}+s_i}{2}) = \frac{\tilde{r}(s_{i+1})-\tilde{r}(s_i)-\tilde{r}(s_{i-1})+\tilde{r}(s_{i-2})}{2\Delta s^2}.$$

This section reformulated a minimum energy path following problem (15) into an SOCP problem, which enables a global optimal solution. Note that this solution is computed by the nominal system model (1). However, there widely exists model uncertainties in practice, and hence the solution based on nominal system model might not yield sufficiently accurate results for given tasks. Therefore, this motivates using a controller to modify the input signal and improve the real time path following performance. Consequently, the next section will introduce an indirect update framework to embed the solution of the SOCP problem. This takes the form of a comprehensive ILC algorithm to enhance path following accuracy with respect to model uncertainties.

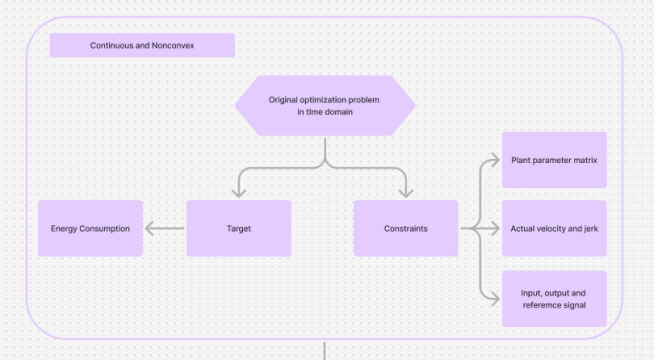


Fig. 1. A visual diagram on the time-spcae coordinate conversion mechanism and the entire algorithm implementation.(The final version is still being revised)

IV. Control Design Approach

This section proposes a comprehensive ILC algorithm with an indirect reference update framework to increase the path following accuracy of industrial robotic tasks. A diagram summarizing the time-space coordinate conversion mechanism and the entire algorithm implementation is shown in Fig. 1. This algorithm is based on the minimum energy path planning strategy and the ILC reference update law. Its robust convergence properties are then discussed.

A. An Indirect Reference Update Framework of ILC

In a wide class of industrial robotic tasks, the robots operate in a repetitive mode. In other words, these robots' end-effectors are required to follow the same path at each trial, and the state q is set to the same initial value at the termination of each trial. It is therefore possible to integrate the technique of ILC with the path planning strategy to increase the practical performance of these robotic path following tasks. In these repetitive tasks, ILC outperforms alternative control strategies, including traditional feedback control and model predictive control, for the following two reasons:

1) It learns the correct input and reference, which does not necessarily need on-line computations at each trial and certainly reduces the computation load.
2) It can reduce the error between successive trials, while the corresponding error of other control methods stays at a similar level for all trials.

To design an ILC law, a subscript k is used to denote the corresponding information at the k^{th} trial. For example, u_k and y_k represent the k^{th} trial's input and output respectively. The spatial error at each trial is

$$\tilde{e}_k = r - \tilde{y}_k^{ex}, \tag{29}$$

where $\tilde{y}_k^{ex}$ denotes the k^{th} trial's measured spatial output.

In the conventional ILC framework, the input signal u_k is updated at each trial to improve the temporal tracking accuracy. However, the input signal u is coupled with the problem (15) as a variable, and the direct input update scheme using ILC breaks the optimization structure. Alternatively, note that the input signal fully depends on the reference profile $\tilde{r}$, as they have a causal relationship linked by the problem (15).

Algorithm 1 Comprehensive ILC Algorithm with Minimum Energy

Input: Reference profile $\tilde{r}$, nominal system model (1)
Output: Optimal input signal u^* and motion profile θ^*

1: **initialization:** Trial number $k = 0$
2: Solve the problem (15) with $\tilde{r}_0 = \tilde{r}$; record the solutions u_0 and θ_0.
3: Apply u_0 to the plant to obtain y_0^{ex}, and transform y_0^{ex} to $\tilde{y}_0^{ex}$ with respect to θ_0; Record $\tilde{e}_0$.
4: **repeat**
5: Update the reference profile using $\tilde{r}_{k+1} = \tilde{r}_k + L\tilde{e}_k$.
6: Set $k \to k+1$.
7: Solve the problem (15) with $\tilde{r}_k$; record the solutions u_k and θ_k.
8: Implement u_k to the plant, and measure $\tilde{y}_k^{ex}$ respect to θ_k; Record $\tilde{e}_k$.
9: **until** $\|\tilde{e}_k\| < \delta$
10: **return** $u^* = u_k$ and $\theta^* = \theta_k$

As a result, instead of applying the conventional input update ILC scheme, an indirect ILC procedure is proposed to update the nominal reference profile, $\tilde{r}_k$, at each trial. Once the nominal reference profile is updated, it is applied to the problem (15), which yields an input signal u_k. This indirect reference update framework, as an approach which alternates optimization of input and reference, gives rise to a practical iterative algorithm (Algorithm 1). In Algorithm 1, the spatial output trajectory $\tilde{y}_k^{ex}$ in Step 3 and 8 should be measured experimentally, $L : L_2^m[0,\ 1] \to L_2^m[0,\ 1]$ is a learning operator and δ is a positive scalar representing the accuracy requirement of particular task.

The requirement of experimentally measured data is not necessary if the nominal system model (1) matches the real plant. However, ILC has a strong track record in achieving high accuracy despite significant modeling error. The desired properties will be further discussed in the next subsection.

Remark 3. In the spatial reference update law $\tilde{r}_{k+1} = \tilde{r}_k + L\tilde{e}_k$, the learning operator L must satisfy the condition $\rho(L) < 1$ to guarantee convergence of the spatial reference iterative update. For simplicity, L is chosen as $L = \gamma I$, where $\gamma \in [0, 1]$ is a scalar. As γ increases, the algorithm converges faster, with the tracking error decreasing concordantly.

B. Robust Convergence Analysis

If nominal system model (1) is accurate, Algorithm 1 provides the optimal solution within a single trial. However, with the existence of model uncertainty, the experimentally measured output y_k^{ex} differs from the numerical computed value y_k even for the same input signal u_k. This algorithm attempts to incorporate real-time measured data (generated by the real plant) in the indirect reference update framework to improve the task performance.

In this section, evidence is provided that Algorithm 1 solves the optimal path following problem with desired error convergence properties despite significant model uncertainty. To

investigate the robust performance, the following assumption is required to describe the model uncertainty.

Assumption 2. Suppose that the difference between measured output y^{ex} and the numerical computed value y, i.e., $\Delta y = y^{ex} - y$, satisfies

$$\|\Delta y\| \leqslant \delta \|y\|, \tag{30}$$

where δ is a non-negative scalar.

This assumption can always be satisfied in practice, as the model uncertainty has a finite range. To analyze the robust convergence properties of Algorithm 1, the next lemma is needed.

Lemma 1. Suppose that the speed profile $\dot{\theta}$ is bounded as

$$\underline{\theta} \leqslant \dot{\theta}(t) \leqslant \overline{\theta}, \ t \in [0, \ T]. \tag{31}$$

The value $\tilde{y}$ is bounded as

$$\sqrt{\underline{\theta}} \|y\| \leqslant \|\tilde{y}\| \leqslant \sqrt{\overline{\theta}} \|y\|. \tag{32}$$

Proof. See Appendix E. □

Using the above lemma, the robust convergence properties of Algorithm 1 are illustrated in the next theorem.

Theorem 1. If the speed profile $\dot{\theta}$ is bounded by (31), and the inequality

$$\|I - L\| + \delta \sqrt{\frac{\overline{\theta}}{\underline{\theta}}} \|L\| < 1 \tag{33}$$

holds, the spatial error $\tilde{e}_k$ generated by Algorithm 1 satisfies the following condition

$$\left\|\tilde{e}'_{k+1}\right\| < \|\tilde{e}_k\|, \tag{34}$$

where $\tilde{e}'_{k+1} = \tilde{r} - (\tilde{y}_{k+1} + \tilde{\Delta y}'_{k+1})$, $y'_{k+1}(t) = \tilde{y}_{k+1}(\theta_k(t))$ and $\tilde{\Delta y}_{k+1}(s) = \Delta y'_{k+1}(\theta_k^{-1}(s))$.

Proof. See Appendix F. □

Theorem 1 illustrates the robust properties of Algorithm 1 between successive trials. The forthcoming theorem elucidates this algorithm's overall robust convergence properties via a randomized coordinate descent methodology [55].

Theorem 2. Suppose that two variables θ and $\tilde{r}$ are updated by the law

$$\begin{cases} \theta_{k+1} = \theta_k^*, & k \text{ is even}, \\ \tilde{r}_{k+1} = \tilde{r}_k + L\tilde{e}_k & k \text{ is odd}, \end{cases} \tag{35}$$

where θ_k^* is the optimizer of problem (15). Then, both the control effort sequence $\{\|u_k\|\}$ and the error sequence $\{\|\tilde{e}_k\|\}$ will converge to their limits respectively, i.e., $\|u\|^*$ and $\|\tilde{e}\|^*$ which is demonstrated in Algorithm 1.

Proof. See Appendix G. □

This theorem delineates the robust convergence properties of Algorithm 1. These properties ensure that both the energy consumption and system error will converge to their limits.

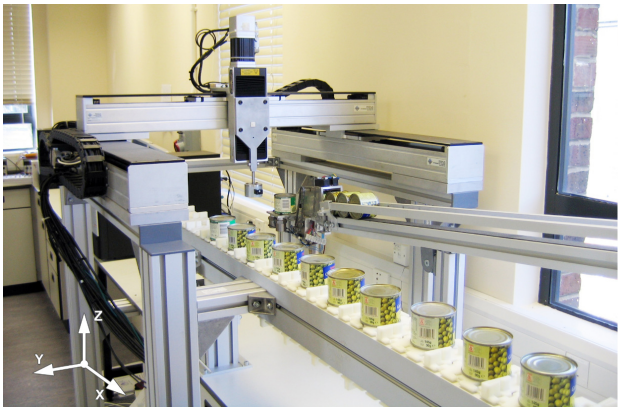


Fig. 2. Three-axis gantry robot test platform.

Based on the results in Theorem 1 and 2, the next corollary further defines some attractive properties.

Corollary 2. Suppose that the system model is linear, then the converged solutions of Algorithm 1 provide the global optimal performance index of the given path following problem (14), i.e., the following two sets of constraints are equivalent as

$$y = G(u), \ \tilde{y} = \tilde{r}^* \Leftrightarrow y = \hat{G}(u), \ \tilde{y} = \tilde{r}, \tag{36}$$

where $\hat{G}$ denotes the accurate system model function.

Proof. See Appendix H. □

This corollary illustrates that the problem solved at each ILC trial will gradually approach the exact problem within certain class of systems, and finally provide the same solution. It can therefore be concluded that, although the real plant model is partially unknown, practical implementation of Algorithm 1 can be carried out to achieve the minimum control energy of the path following task, and achieve path following accuracy at the same time.

V. Verification on a Gantry Robot

This section employs a three-axis gantry robot as a test platform to verify the practical performance of Algorithm 1.

A. Test Platform Description

The gantry robot shown in Fig. 2 replicates the manufacturing environment including model uncertainty and random disturbance. The three axes (x, y and z) shown in the figure are perpendicular to each other, and the hybrid motion of three axes yields the 3D space motion of its end-effector in a given region. The gantry robot performs manufacturing tasks such as picking up a payload from the dispenser and placing it down on a moving convey situated below.

The input of the gantry robot has units of voltage, and the displacement of each axis is measured by an incremental encoder as the output. The resolutions of the encoders on x-axis, y-axis and z-axis are $32\mu m$, $32\mu m$ and $16\mu m$ respectively. The gantry robot is controlled by a powerful dSPACE device DS1103. See Fig. 3 for a block structure overview of the gantry robot test platform.

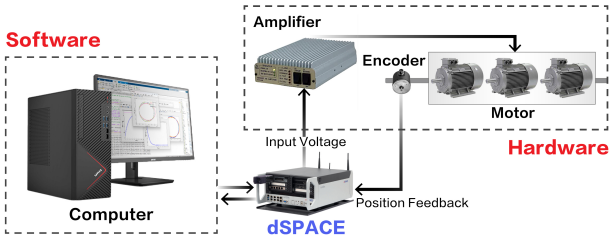


Fig. 3. Gantry robot block structure overview.

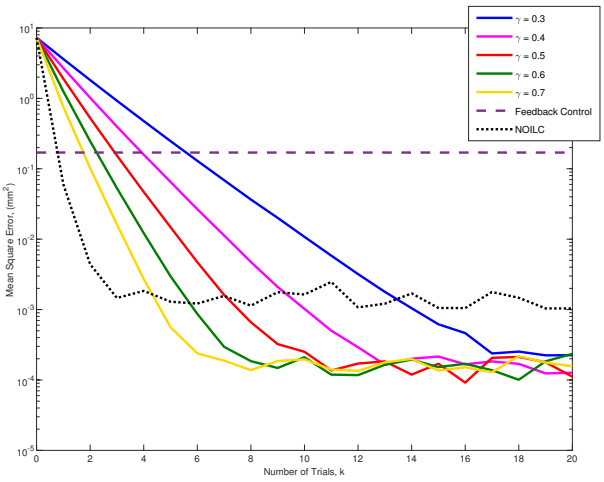


Fig. 4. Mean square path following error on each trial for $\gamma = 0.3$, 0.4, 0.5, 0.6 and 0.7 compared with NOILC and proportional feedback control.

B. Task Specifications

The objective of the path following task is to use the x-axis and z-axis ($m = 2$) of the gantry robot to repetitively follow a 2D circular path (shown as the yellow curve in Fig. 5) defined as

$$\tilde{r}(s) = \begin{bmatrix} 0.005\cos(-2\pi s + \pi) + 0.005 \\ 0.005\sin(-2\pi s + \pi) \end{bmatrix}, \; s \in [0,\ 1] \quad (37)$$

during a given time period $T = 2$ seconds. The nominal system models of x-axis and z-axis are given as the following transfer function representations

$$G_x(s) = \frac{0.05}{s} \text{ and } G_z(s) = \frac{0.03}{s}, \quad (38)$$

and each axis is in series with an integrator. An input saturation constraint

$$|\tilde{u}(s)| \preceq \begin{bmatrix} 1 \\ 2.5 \end{bmatrix}, \; s \in [0,\ 1] \quad (39)$$

is also considered to prevent the damage to gantry robot, and added into system constraint set $\mathcal{C}$.

C. Experimental Results

Algorithm 1 is implemented on the gantry robot with $\gamma = 0.3$, 0.4, 0.5, 0.6 and 0.7 to perform the repetitive path following task described in the previous subsection. The mean square path following error on each trial is plotted in Fig. 4, for different values of γ. The error convergence rate is proportional to the value of γ. Also, due to the incremental encoder resolutions, the errors all converge to around 10^{-4}, which illustrates the ability of this algorithm to increase the path following accuracy.

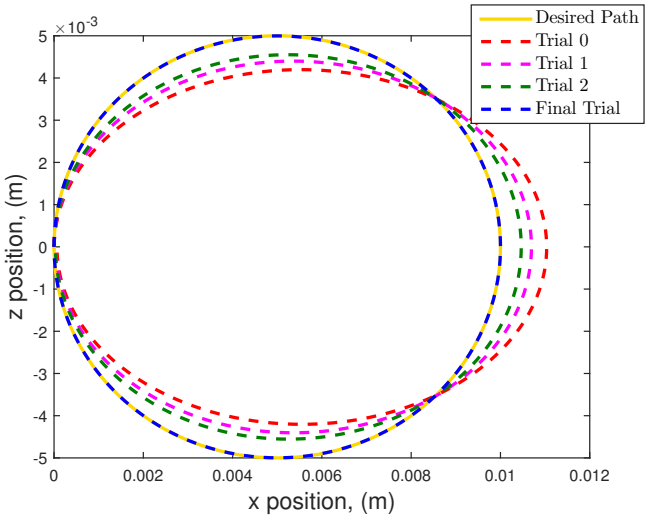


Fig. 5. Comparison between desired path and the hybrid output trajectories over the first three trials and final trial for $\gamma = 0.3$.

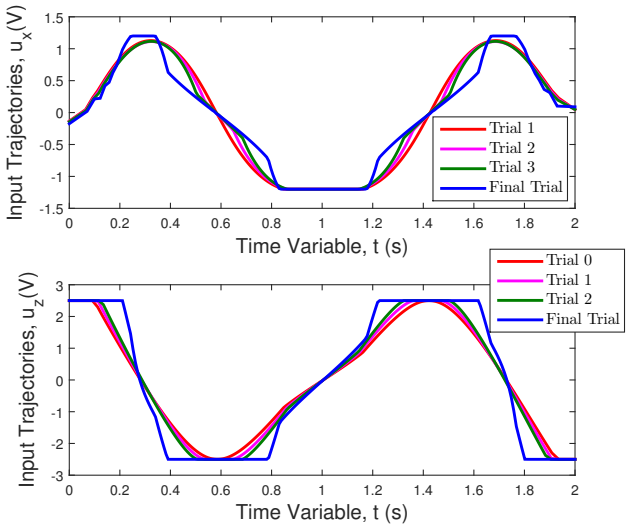


Fig. 6. Input trajectories over first three trials and on the final trial for $\gamma = 0.3$.

The results with $\gamma = 0.3$ are selected for further analysis. The measured hybrid output trajectories over the first three trials and on the final trial are plotted in Fig. 5 as the dashed curves, and the given reference profile (37) is also plotted as the yellow curve for comparison. From this figure, note that there exists a certain level of mismatch between the output and the given path for first few trials, which is clearly beyond the scope of practical tolerance for path following error. However, the output trajectory of the final trial accurately follows the desired path. This fact again confirms that this algorithm improves the path following accuracy using ILC reference

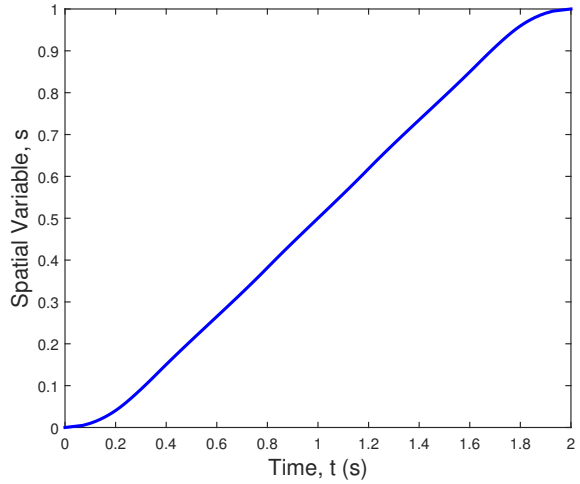


Fig. 7. Converged motion profile for $\gamma = 0.3$.

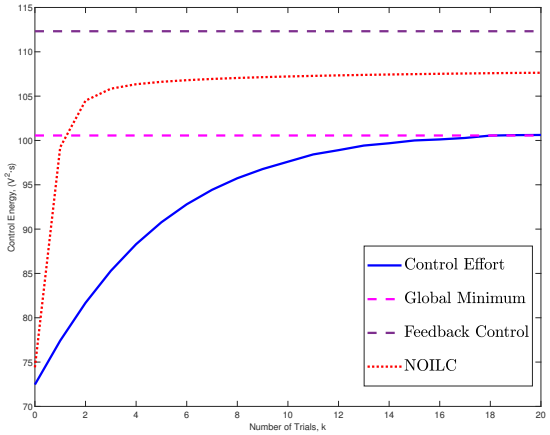


Fig. 8. Control energy compared with the minimum value, and with the energy expended using NOILC proportional feedback control.

update law.

The input trajectories at each trial for $\gamma = 0.3$, are shown in Fig. 6. Note that the input gradually converges to the final one, and all inputs stay within the input saturation constraint set (39), which verifies the feasibility of this algorithm to handle system constraints. In addition, the final optimal motion profile is plotted in Fig. 7. This solution suggests that the best way to save control energy is to gradually accelerate at the beginning, then keep an almost constant speed, and finally gradually decelerate at the trial end.

Moreover, the control energy on each trial for $\gamma = 0.3$ is plotted as the blue curve in Fig. 8. For comparison, the identified system model of the gantry robot is considered, i.e.,

$$\hat{G}_x(s) = \frac{1.67 \times 10^{-5}(s + 500.2)(s + 4.9 \times 10^5)...}{s(s^2 + 24s + 6401)...} \frac{(s^2 + 10.58s + 1.145 \times 10^4)(s^2 + 21.98s + 2.9 \times 10^4)}{(s^2 + 21.38s + 2.017 \times 10^4)(s^2 + 139.5s + 2.162 \times 10^5)}$$
$$\text{and } \hat{G}_z(s) = \frac{15.8869(s + 850.3)}{s(s^2 + 707.6s + 3.377 \times 10^5)} \quad (40)$$

obtained using frequency response fitting, with full details provided in [56]. This model is applied to the minimum energy path following problem (15) to yield the solutions for input and motion profile. As this identified model is accurate, there is no need to perform the reference update procedure, and the resulting control energy is regarded as the global minimum, which is plotted in the same figure as the dashed magenta line. Note that the blue curve converges to the dashed magenta line. The reason is that the gantry robot consists of stepper motors, whose system model is approximately equivalent to an integrator. As each axis is in series with an integrator, the overall system can be considered as a double integrator system, whose model uncertainty is obviously time invariant and independent of the motion profile. Then, according to Corollary 2, the converged solutions of Algorithm 1 provide the global minimum control energy of the given path following problem in the case of a linear system.

For comparison, the feedback control method and NOILC algorithm are also applied to the same path following task with constant motion profile. The corresponding path following error and control energy consumption are plotted as the dashed lines in Fig. 4 and Fig. 8 respectively. From the comparison, it is obvious that the path following error using feedback control is around $1,000$ times larger and NOILC is around 10 times larger than the error of Algorithm 1. In addition, an 11.69% energy reduction is achieved using the path planning strategy of Algorithm 1 compared to the energy required by feedback control and a 6.97% energy reduction is achieved compared to NOILC. Therefore, Algorithm 1 outperforms the classical control method both in terms of path following accuracy and energy reduction.

D. Case Study

In this subsection, a simulation involving a 3-degree-of-freedom (3-DOF) serial robot manipulator is conducted to validate the effectiveness of Algorithm 1 on highly nonlinear systems.

The objective of the path following task is to use the 3-DOF serial robot manipulator ($m = 3$) to repetitively follow a 3D circular path (shown as the yellow curve in Fig. 9) defined as

$$\tilde{r}(s) = \begin{bmatrix} 0.005\cos(-6\pi s + \pi) + 0.005 \\ 0.005\sin(-6\pi s + \pi) \\ 0.05s \end{bmatrix}, \ s \in [0,\ 1] \quad (41)$$

over a time period of $T = 2$ seconds. The nominal system model of the serial robot manipulator is given as plant (1),

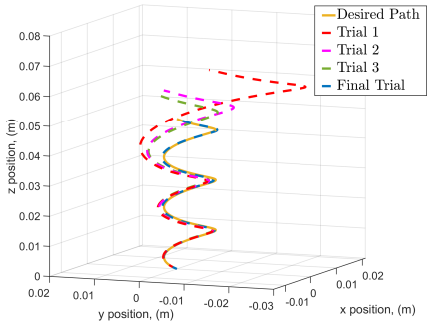


Fig. 9. Comparison between desired path and the hybrid output trajectories of the 3-DOF serial robot manipulator over the first three trials and on the final trial for $\gamma = 0.3$.

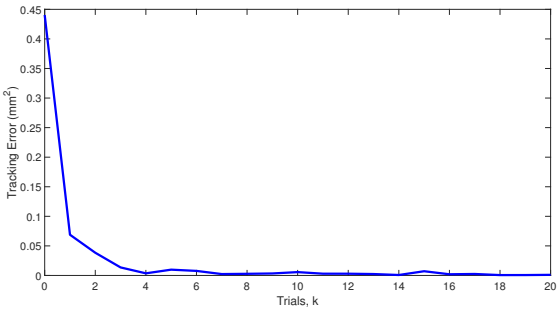


Fig. 10. Mean square path following error of 3-DOF serial robot manipulator on each trial for $\gamma = 0.3$.

where the specific state and parameter matrix representation is

$$q = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^\top, \quad R = I_3,$$

$$M = \begin{bmatrix} 4.5 & 1.25\cos(q_2) & 0.4\cos(q_3) \\ 1.25\cos(q_2) & 2.5 & 0.4\cos(q_3) \\ 0.4\cos(q_3) & 0.4\cos(q_3) & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & -1.25\sin(q_2)\dot{q}_2 & -0.4\sin(q_3)\dot{q}_3 \\ 1.25\sin(q_2)\dot{q}_1 & 0 & -0.4\sin(q_3)\dot{q}_3 \\ 0.4\sin(q_3)\dot{q}_1 & 0.4\sin(q_3)\dot{q}_1 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 9.81(\cos q_1 + 1.2\cos(q_1+q_2) + 0.6\cos(q_1+q_2+q_3)) \\ 9.81(0.6\cos(q_1+q_2) + 0.3\cos(q_1+q_2+q_3)) \\ 2.943\cos(q_1+q_2+q_3) \end{bmatrix}$$

Algorithm 1 is implemented on the robot manipulator with $\gamma = 0.3$. The mean square path following error on each trial is plotted in Fig. 10 and the control energy on each trial is plotted in Fig. 11.

As seen from Fig. 10, despite the high nonlinearity of the model, the error of Algorithm 1 can still converge. However, considering that the discretization method will cause distortion of the nonlinear signal, the error cannot achieve a monotonic decrease. Fig. 11 shows the initial energy consumption does not start from 0, just as shown in Proposition 2. As long as the system is in the startup state, the robot arm needs to compensate for the influence of gravity against it. In particular, there is always an input $\hat{u}$ to resist gravity, and there will exist energy consumption $\hat{E}$.

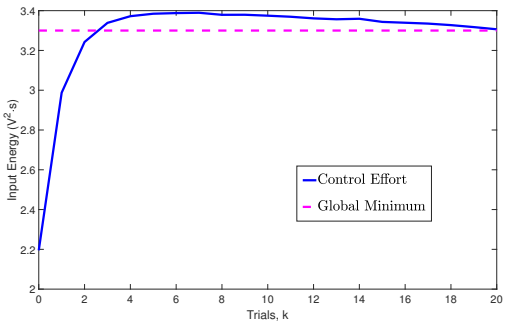


Fig. 11. Control energy of 3-DOF serial robot manipulator comparison with the minimum value for $\gamma = 0.3$.

VI. Conclusion and Future Work

This paper provides a comprehensive algorithm for robots in manufacture, which achieves energy reduction and high path following accuracy at the same time. A path following problem with system constraints is formulated to minimize the control energy. This problem is reformulated in spatial domain as a convex optimization problem, which is solved by a discretized approach. Then, a comprehensive ILC algorithm is proposed via an indirect reference update framework to increase the path following accuracy under model uncertainties, which combines a minimum energy path planning strategy and a reference profile update precedure. The experimental results on a gantry robot demonstrate its feasibility, and the comparison with unplanned motion profiles verifies its advantages and novelty.

Future work will focus on designing the learning operator L to improve algorithm performance and formulating performance indices to quantify the effect of disturbance inputs. In addition, the problem setup will be expanded to handle the energy problem for an even more general class of system.

Appendix A
Proof of Proposition 1

According to the definitions in (12), it is clear that

$$b(s) \geqslant 0, \tag{42}$$

and the two functions $a(s)$ and $b(s)$ have

$$\dot{b}(s) = b^{'}(s)\dot{s} = \frac{\mathrm{d}(\dot{s}^2)}{\mathrm{d}t} = 2\ddot{s}\dot{s} = 2a(s)\dot{s},$$

which is equivalent to

$$b^{'}(s) = 2a(s). \tag{43}$$

Furthermore, the initial and terminal condition of $b(s)$ is

$$b(0) = \dot{s}^2(0) = 0, \ b(1) = \dot{s}^2(\hat{T}) = 0, \tag{44}$$

which respects to the fact that the end-effector starts from its initial position and stopped at the terminal position.

In addition, the overall tracking time $\hat{T}$ is represented in spatial domain by

$$\hat{T} = \int_0^{\hat{T}} 1\mathrm{d}t = \int_{\theta(0)}^{\theta(\hat{T})} \dot{s}^{-1}\mathrm{d}s = \int_0^1 \frac{1}{\sqrt{b(s)}}\mathrm{d}s, \tag{45}$$

and similarly the cost function $\|u\|^2$ can be represented in spatial domain by

$$\begin{aligned}\|u\|^2 &= \int_0^T u^\top(t)u(t)\mathrm{d}t = \int_0^{\hat{T}} u^\top(t)u(t)\mathrm{d}t + \hat{E}(\hat{T},\hat{u}) \\ &= \int_{\theta(0)}^{\theta(\hat{T})} \tilde{u}^\top(s)\tilde{u}(s)\dot{s}^{-1}\mathrm{d}s + \hat{E}(\hat{T},\hat{u}) \\ &= \int_0^1 \frac{\tilde{u}^\top(s)\tilde{u}(s)}{\sqrt{b(s)}}\mathrm{d}s + \hat{E}(\hat{T},\hat{u}),\end{aligned} \tag{46}$$

where $\hat{E}(\hat{T},\hat{u}) = (T-\hat{T})\hat{u}^\top\hat{u}$ is a term dependent on $\hat{T}$ and $\hat{u}$, representing the energy consumption during the interval $[\hat{T},\ T]$.

Using these spatial notations, the problem (15) can be reformulated. The problem (20) involves three variables u, a and b. It follows from [57] that the function x^2/y is convex respect to (x,y) for $y \geqslant 0$, and integration reserves the convexity, so the cost function is convex. Due to the same reason, the left side function in the time constraint (20e) is also convex, so this inequality constraint is convex. Since the dynamic constraint (20b) is linear and affine in u, a and b, the relationship (20c) between a and b is linear. As the derivative is a linear operator, the inequality constraint (20d) is convex. Also, the system constraint (20f) is convex as the set $\mathcal{C}$ is convex. Therefore, problem (20) is a convex optimization problem.

APPENDIX B
PROOF OF PROPOSITION 2

To minimize the energy consumption $E(\hat{T},\hat{u})$ subject to the motion termination constraint $R(\hat{q})u(t) = D(\hat{q})$, we formulate the optimization problem as follows:

$$\min_{\hat{u}} \hat{E}(\hat{T},\hat{u}) \tag{47a}$$

$$\text{s.t. } R(\hat{q})\hat{u} = D(\hat{q}) \tag{47b}$$

According to the Lagrange multiplier method, for any given $\hat{T}$, we can transform the original optimization problem (47) into finding the extremum of the Lagrangian

$$f(\hat{u},\lambda) = \hat{u}^\top\hat{u} - \lambda^\top(R(\hat{q})\hat{u} - D(\hat{q})), \tag{48}$$

where λ is the Lagrange multiplier. By solving this, we obtain:

$$\lambda = 2(R(\hat{q})R^\top(\hat{q}))^{-1}D(\hat{q}), \tag{49}$$

$$\hat{u} = R^\top(\hat{q})(R(\hat{q})R^\top(\hat{q}))^{-1}D(\hat{q}), \tag{50}$$

which is identical in form to the least-squares solution of the equation $R(\hat{q})u(t) = D(\hat{q})$, thus completing the proof.

APPENDIX C
PROOF OF PROPOSITION 3

From Assumption 1, the value of $b^{'}(s)$ on this sub-interval is also constant by relationship (20c), and it follows that

$$\frac{b_i - b_{i-1}}{s_i - s_{i-1}} = b^{'}(s) = 2a_i,\ i = 1,\ldots,N, \tag{51}$$

which gives rise to

$$b_i - b_{i-1} = 2a_i\cdot\Delta s,\ i = 1,\ldots,N, \tag{52}$$

and

$$b(s) = b_i + (s - s_i)\cdot\frac{b_{i+1} - b_i}{s_{i+1} - s_i},\ s \in [s_i,\ s_{i+1}]. \tag{53}$$

The discretized form of the cost function (20a) is

$$\begin{aligned}\int_0^1 \frac{\tilde{u}^\top(s)\tilde{u}(s)}{\sqrt{b(s)}}\mathrm{d}s &= \sum_{i=1}^N \int_{s_{i-1}}^{s_i} \frac{\tilde{u}^\top(s)\tilde{u}(s)}{\sqrt{b(s)}}\mathrm{d}s \\ &= \sum_{i=1}^N \int_{s_{i-1}}^{s_i} u_i^\top u_i\cdot(b_{i-1} + (s - s_{i-1})\cdot\frac{b_i - b_{i-1}}{s_i - s_{i-1}})^{-\frac{1}{2}}\mathrm{d}s \\ &= \sum_{i=1}^N u_i^\top u_i \frac{2(s_i - s_{i-1})}{b_i - b_{i-1}}(b_{i-1} + (s - s_{i-1})\frac{b_i - b_{i-1}}{s_i - s_{i-1}})^{\frac{1}{2}}|_{s_{i-1}}^{s_i} \\ &= \sum_{i=1}^N u_i^\top u_i \frac{2(s_i - s_{i-1})}{b_i - b_{i-1}}(b_i - b_{i-1})^{\frac{1}{2}} \\ &= \sum_{i=1}^N \frac{2\Delta s\cdot u_i^\top u_i}{\sqrt{b_{i-1}} + \sqrt{b_i}},\end{aligned} \tag{54}$$

and the discretized form for time constraint (20e) is

$$\sum_{i=1}^N \frac{2\Delta s}{\sqrt{b_{i-1}} + \sqrt{b_i}} \leqslant T. \tag{55}$$

Furthermore, the dynamic constraint (20b) can be represented in discretized form as

$$\tilde{R}_i u_i = \tilde{M}_i a_i + \tilde{C}_i\cdot\frac{b_{i-1} + b_i}{2} + \tilde{d}_i,\ i = 1,\ldots,N. \tag{56}$$

In addition, the system constraint (20f) is converted into its discretized form (26f). Hence, all above steps yield the discretized form (26).

APPENDIX D
PROOF OF COROLLARY 1

To rewrite (26) into a SOCP problem with a standard form, three extra variables are first introduced as

$$c = [c_0,\ldots,c_N]^\top,\ d = [d_1,\ldots,d_N]^\top,\ \hat{d} = [\hat{d}_1,\ldots,\hat{d}_N]^\top.$$

The cost function (26a), and the time constraint (26e) can be equivalently written as a linear forms as

$$\sum_{i=1}^N 2\Delta s\cdot d_i,\ \sum_{i=1}^N 2\Delta s\cdot\hat{d}_i = T \tag{57}$$

with the inequality constraints

$$\frac{u_i^\top u_i}{\sqrt{b_{i-1}} + \sqrt{b_i}} \leqslant d_i,\ \frac{1}{\sqrt{b_{i-1}} + \sqrt{b_i}} \leqslant \hat{d}_i,\ i = 1,\ldots,N, \tag{58}$$

which can be further replaced as

$$\begin{aligned}&\frac{u_i^\top u_i}{c_{i-1} + c_i} \leqslant d_i,\ \frac{1}{c_{i-1} + c_i} \leqslant \hat{d}_i,\ i = 1,\ldots,N, \\ &c_i \leqslant \sqrt{b_i},\ i = 0,\ldots,N.\end{aligned} \tag{59}$$

The inequality (59) is written as three second order cone constraints as follows:

$$\begin{aligned}\left\|\begin{matrix}2u_i\\ c_{i-1}+c_i-d_i\end{matrix}\right\|_2 &\leqslant c_{i-1}+c_i+d_i,\ i=1,\ldots,N,\\ \left\|\begin{matrix}2\\ c_{i-1}+c_i-\hat{d}_i\end{matrix}\right\|_2 &\leqslant c_{i-1}+c_i+\hat{d}_i,\ i=1,\ldots,N,\\ \left\|\begin{matrix}2c_i\\ b_i-1\end{matrix}\right\|_2 &\leqslant b_i+1,\ i=0,\ldots,N. \end{aligned} \tag{60}$$

Hence, the above steps generate the SOCP problem (28).

APPENDIX E
PROOF OF LEMMA 1

The $2-$norm in spatial domain is equivalently written as

$$\begin{aligned}\|\tilde{y}\|^2 &= \int_0^1 \tilde{y}^\top(s)\tilde{y}(s)\mathrm{d}s\\ &= \int_0^{\hat{T}} \dot{\theta}(t)y^\top(t)y(t)\mathrm{d}t \end{aligned} \tag{61}$$

in the time domain. As the speed profile $\dot{\theta}$ is bounded as shown in (31), it follows that

$$\|\tilde{y}\|^2 \leqslant \overline{\theta}\int_0^{\hat{T}} y^\top(t)y(t)\mathrm{d}t = \overline{\theta}\|y\|^2, \tag{62}$$

and in symmetry, we can derive the lower bound, which together give rise to the bounded condition (32).

APPENDIX F
PROOF OF THEOREM 1

The spatial error equation (29) is equivalent to

$$\tilde{e}_k = \tilde{r} - \tilde{y}_k^{ex} = \tilde{r} - (\tilde{y}_k + \tilde{\Delta y}_k), \tag{63}$$

which together with definition of $\tilde{e}_k^{'}$ gives rise to

$$\tilde{e}_{k+1}^{'} - \tilde{e}_k = (\tilde{y}_k - \tilde{y}_{k+1}) + (\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'}). \tag{64}$$

Due to the path following requirement, there exists $\tilde{y}_k = \tilde{r}_k$, which together with the ILC law yield

$$\tilde{e}_{k+1}^{'} = (I-L)\tilde{e}_k + (\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'}), \tag{65}$$

and has a norm inequality

$$\left\|\tilde{e}_{k+1}^{'}\right\| \leqslant \|I-L\|\|\tilde{e}_k\| + \left\|(\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'})\right\|. \tag{66}$$

Since the speed profile $\dot{\theta}$ is bounded by (31), it follows from Lemma 1 that

$$\left\|\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'}\right\| \leqslant \sqrt{\overline{\theta}}\left\|\Delta y_k - \Delta y_{k+1}^{'}\right\|, \tag{67}$$

$$\left\|y_k - y_{k+1}^{'}\right\| \leqslant \frac{1}{\sqrt{\underline{\theta}}}\|\tilde{y}_k - \tilde{y}_{k+1}\|. \tag{68}$$

According to the model uncertainty bounded condition (32) in Assumption 2, there exists

$$\left\|\Delta y_k - \Delta y_{k+1}^{'}\right\| \leqslant \delta\left\|y_k - y_{k+1}^{'}\right\|. \tag{69}$$

The above inequalitues (67), (68) and (69) yield

$$\left\|\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'}\right\| \leqslant \delta\sqrt{\frac{\overline{\theta}}{\underline{\theta}}}\|\tilde{y}_k - \tilde{y}_{k+1}\|. \tag{70}$$

which further implies

$$\left\|\tilde{\Delta y}_k - \tilde{\Delta y}_{k+1}^{'}\right\| \leqslant \delta\sqrt{\frac{\overline{\theta}}{\underline{\theta}}}\|L\|\|\tilde{e}_k\|. \tag{71}$$

Substitute (71) into (66) to give

$$\left\|\tilde{e}_{k+1}^{'}\right\| \leqslant (\|I-L\| + \delta\sqrt{\frac{\overline{\theta}}{\underline{\theta}}}\|L\|)\|\tilde{e}_k\|. \tag{72}$$

Therefore, if the condition (33) holds, the error reduction property (34) holds.

APPENDIX G
PROOF OF THEOREM 2

In randomized coordinate descent algorithm, the update component is chosen randomly at each iteration. Before proving Theorem 2, the convergence lemma of randomized coordinate descent algorithm will be introduced first.

Lemma 2. [58] Suppose a function $f:\mathbb{R}^n \mapsto \mathbb{R}$ is strictly convex satisfying

$$f(y) > f(x) + \nabla f(x)^\top(y-x) + \frac{\sigma}{2}\|y-x\|^2, \tag{73}$$

where $\sigma > 0$ denotes the modulus of convexity. Then, for all trial numbers k, there exists

$$E(f(x_k)) - f^* \leq (1 - \frac{\sigma}{nL_{\max}})^k(f(x_0) - f^*), \tag{74}$$

where operator $E(\cdot)$ denotes the expectation with respect to all random variables of f, and $L_{\max}$ is the maximum component Lipschitz constant.

According to the definition of spatial error $\tilde{e}$, the cost function (15a) can be equivalently converted to

$$\begin{aligned}&\min_{\theta,\tilde{r}} J(\theta,\tilde{r}),\\ &J(\theta,\tilde{r}) = \|\tilde{e}\|^2 + \alpha\|u\|^2, \end{aligned} \tag{75}$$

in a Lagrange multiplier form, where α is the Lagrange multiplier with respect to (15c) and two variables θ and $\tilde{r}$ are viewed as two coordinates of (75). Based on Lemma 2 and Corollary 1, the above equivalent problem can be solved with the update law (35). When k is odd, the cost function (75) decreases along the direction of $\nabla_{\tilde{r}_k}J$, and when k is even, the whole course of searching the optimizer of (15) is equivalent to multiply randomized descents along the direction of $\nabla_{\theta_k}J$. Thus, according to (74), the equivalent cost function $J(\theta,\tilde{r})$ will converge to a theoretical minimum J^*.

As Theorem 1 illustrates the bounded monotonical property of spatial error between adjacent trials, along with the converge of $J(\theta,\tilde{r})$, both $\{\|u_k\|\}$ and $\{\|\tilde{e}_k\|\}$ will converge to their limit $\|u\|^*$ and $\|\tilde{e}\|^*$ respectively.

APPENDIX H
PROOF OF COROLLARY 2

As the system is linear, the difference Δy of model uncertainty satisfies

$$\Delta y = Hy, \tag{76}$$

where $H : L_2^m[0,\ T] \mapsto L_2^m[0,\ T]$ is an operator mapping y to Δy.

While the path following error converges to zero, and there exists

$$\tilde{y}^{ex*} = \tilde{r}. \tag{77}$$

where the superscript $*$ indicates the optim Also, according to the path following task requirement, the following condition

$$\tilde{y}^* = \tilde{r}^* \tag{78}$$

holds. Assume that there is an invertible map defined as $\Gamma : f(t) \mapsto \tilde{f}(s)$, where $f(t)$ is a class of time function and $\tilde{f}(s)$ is a class of spatial variable function. Then, it follows from the model uncertainty condition (76) such that

$$\tilde{r} = \tilde{y}^{ex*} = \Gamma((I+H)y^*) = \Gamma((I+H)r^*). \tag{79}$$

The tracking constraints for optimal path planning problem (14) at the corresponding trial is

$$y = G(u),\ \tilde{y} = \tilde{r}^*, \tag{80}$$

which yields

$$\begin{aligned} y^{ex} &= (I+H)y = (I+H)G(u) = \hat{G}(u), \\ \tilde{y}^{ex} &= \Gamma((I+H)y) = \Gamma((I+H)\Gamma^{-1}(\tilde{y})) \\ &= \Gamma((I+H)\Gamma^{-1}(\tilde{r}^*)) = \Gamma((I+H)r^*) = \tilde{r}. \end{aligned} \tag{81}$$

The above equivalent constraint is exactly the same as that of the optimal problem using the accurate system model $\hat{G}$ and the actual reference profile $\tilde{r}$, i.e.,

$$y = \hat{G}(u),\ \tilde{y} = \tilde{r}. \tag{82}$$

Therefore, the problem solved at the final trial is equivalent to the problem in the ideal case, and provides the global optimal performance index.

REFERENCES

[1] T. Brogardh, "Present and future robot control development -an industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007.

[2] W. Wang, W. Ding, C. Hua, H. Zhang, H. Feng, and Y. Yao, "A digital twin for 3D path planning of large-span curved-arm gantry robot," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102330, 2022.

[3] X. Zhou, X. Wang, Z. Xie, F. Li, and X. Gu, "Online obstacle avoidance path planning and application for arc welding robot," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102413, 2022.

[4] J. Wang, X. Xi, Y. Zhang, L. Qin, Y. Liu, and W. Zhao, "Path optimization for multi-axis edm drilling of combustor liner cooling holes using SCGA algorithm," *Computers and Industrial Engineering*, vol. 157, p. 107319, 2021.

[5] K. Ly, J. V. Mayekar, S. Aguasvivas, C. Keplinger, M. E. Rentschler, and N. Correll, "Electro-hydraulic rolling soft wheel: Design, hybrid dynamic modeling, and model predictive control," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3044–3063, 2022.

[6] K. Shi, C. Liu, Z. Sun, and X. Yue, "Coupled orbit-attitude dynamics and trajectory tracking control for spacecraft electromagnetic docking," *Applied Mathematical Modelling*, vol. 101, pp. 553–572, 2022.

[7] H. Chen, J. Yang, and H. Ding, "Robotic grinding of curved parts with two degrees of freedom active compliant force-controlled end-effector using decoupling control algorithm," *Robotics and Computer-Integrated Manufacturing*, vol. 93, p. 102935, 2025.

[8] Y. Chen, B. Chu, and C. T. Freeman, "Iterative learning control for path-following tasks with performance optimization," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 234–246, 2021.

[9] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.

[10] E. Barnett and C. Gosselin, "A bisection algorithm for time-optimal trajectory planning along fully specified paths," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 131–145, 2020.

[11] M. Chen and D. Zhu, "Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 401–14 412, 2020.

[12] S. Lin, C. Hu, S. He, W. Zhao, Z. Wang, and Y. Zhu, "Real-time local greedy search for multiaxis globally time-optimal trajectory," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 2, pp. 960–971, 2022.

[13] M. Faroni, A. Umbrico, M. Beschi, A. Orlandini, A. Cesta, and N. Pedrocchi, "Optimal task and motion planning and execution for multiagent systems in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 54, no. 6, pp. 3366–3377, 2024.

[14] Y. Ning, M. Yue, J. Shangguan, and J. Zhao, "Optimal trajectory planning method for the navigation of wip vehicles in unknown environments: Theory and experiment," *IEEE Transactions on Cybernetics*, vol. 53, no. 10, pp. 6317–6328, 2023.

[15] H. Chen and X. Zhang, "Path planning for intelligent vehicle collision avoidance of dynamic pedestrian using Att-LSTM, MSFM, and MPC at unsignalized crosswalk," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4285–4295, 2022.

[16] Y. Xu, L. Liu, N. Gu, D. Wang, and Z. Peng, "Multi-ASV collision avoidance for point-to-point transitions based on heading-constrained control barrier functions with experiment," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 6, pp. 1494–1497, 2023.

[17] Y. Shi, Z. Zhang, L. Xie, and H. Su, "ILC-based two-layer strategy for economic performance improvement in industrial MPC systems," *Journal of Process Control*, vol. 108, pp. 136–147, 2021.

[18] M. van de Vosse, T. W. Toner, M. J. Wu, D. M. Tilbury, and K. L. Barton, "Using economic iterative learning control for time-optimal control of a redundant manipulator," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2023, pp. 1–7.

[19] C. Zhang, H.-K. Lam, J. Qiu, P. Qi, and Q. Chen, "Fuzzy-model-based output feedback steering control in autonomous driving subject to actuator constraints," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 457–470, 2021.

[20] S. You, G. Kim, S. Lee, D. Shin, and W. Kim, "Neural approximation based adaptive control using reinforced gain for steering wheel torque tracking of electric power steering system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 7, pp. 4216–4225, 2023.

[21] Y. Chen, X. Zhang, and J. Wang, "Robust vehicle driver assistance control for handover scenarios considering driving performances," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 7, pp. 4160–4170, 2021.

[22] M. Deyuan and Z. Jingyao, "Robust optimization-based iterative learning control for nonlinear systems with nonrepetitive uncertainties," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 1001–1014, 2021.

[23] D. Shen and J.-X. Xu, "An iterative learning control algorithm with gain adaptation for stochastic systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1280–1287, 2019.

[24] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–144, 2006.

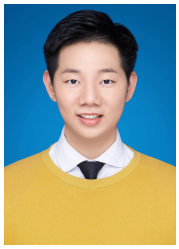
[25] N. Strijbosch and T. Oomen, "Iterative learning control for intermittently sampled data: Monotonic convergence, design, and applications," *Automatica*, vol. 139, p. 110171, 2022.

[26] R. Chi, H. Li, N. Lin, and B. Huang, "Data-driven indirect iterative learning control," *IEEE Transactions on Cybernetics*, 2023.

[27] Q. Wang, S. Jin, and Z. Hou, "Event-triggered cooperative model-free adaptive iterative learning control for multiple subway trains with actuator faults," *IEEE Transactions on Cybernetics*, 2023.

[28] Y. Chen, B. Chu, and C. T. Freeman, "Generalized iterative learning control using successive projection: Algorithm, convergence, and experimental verification," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2079–2091, 2019.
[29] L. Yang, Y. Li, D. Huang, J. Xia, and X. Zhou, "Spatial iterative learning control for robotic path learning," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5789–5798, 2022.
[30] P. T. Kyaw, A. V. Le, P. Veerajagadheswar, M. R. Elara, T. T. Thu, N. H. K. Nhan, P. Van Duc, and M. B. Vu, "Energy-efficient path planning of reconfigurable robots in complex environments," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2481–2494, 2022.
[31] J. Luo, J. Zhuang, M. Jin, F. Xu, and Y. Su, "An energy-efficient path planning method for unmanned surface vehicle in a time-variant maritime environment," *Ocean Engineering*, vol. 301, p. 117544, 2024.
[32] R.-C. Roman, R.-E. Precup, E.-L. Hedrea, S. Preitl, I. A. Zamfirache, C.-A. Bojan-Dragos, and E. M. Petriu, "Iterative feedback tuning algorithm for tower crane systems," *Procedia Computer Science*, vol. 199, pp. 157–165, 2022.
[33] M. Zhang, Y. Cao, J. Huang, and X. Chen, "Cross-backstepping control with prescribed performance for input-coupled underactuated systems under arbitrary initial conditions," *Journal of the Franklin Institute*, vol. 360, no. 16, pp. 11 892–11 915, 2023.
[34] R.-C. Roman, R.-E. Precup, E. M. Petriu, and A.-I. Borlea, "Hybrid data-driven active disturbance rejection sliding mode control with tower crane systems validation," *Sci. Technol*, vol. 27, pp. 3–17, 2024.
[35] I. A. Zamfirache, R.-E. Precup, and E. M. Petriu, "Q-learning, policy iteration and actor-critic reinforcement learning combined with metaheuristic algorithms in servo system control," *Facta Universitatis, Series: Mechanical Engineering*, vol. 21, no. 4, pp. 615–630, 2023.
[36] Y. Wang, X. Zhang, Z. Li, X. Chen, and C.-Y. Su, "Adaptive implicit inverse control for a class of butterfly-like hysteretic nonlinear systems and its application to dielectric elastomer actuators," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 731–740, 2022.
[37] R.-E. Precup, R.-C. Roman, and A. Safaei, *Data-driven model-free controllers*. CRC Press, 2021.
[38] J. Xu, D. Li, and J. Zhang, "Extended state observer based dynamic iterative learning for trajectory tracking control of a six-degrees-of-freedom manipulator," *ISA Transactions*, vol. 143, pp. 630–646, 2023.
[39] J. A. Drallmeier, J. B. Siegel, and A. G. Stefanopoulou, "Iterative learning-based trajectory optimization using fourier series basis functions," *IEEE Control Systems Letters*, vol. 6, pp. 2180–2185, 2022.
[40] Y. Chen and C. T. Freeman, "Iterative learning control for piecewise arc path tracking with validation on a gantry robot manufacturing platform," *ISA Transactions*, vol. 139, pp. 650–659, 2023.
[41] S.-L. Chen, S.-M. Hsieh, and T.-Q. Ta, "Iterative learning contouring control for five-axis machine tools and industrial robots," *Mechatronics*, vol. 94, p. 103030, 2023.
[42] G. Liu and Z. Hou, "Adaptive iterative learning fault-tolerant control for state constrained nonlinear systems with randomly varying iteration lengths," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[43] R. Yang, Y. Gong, and W. Paszke, "ILC-based tracking control for linear systems with external disturbances via an SMC scheme," *IEEE Transactions on Automation Science and Engineering*, 2024.
[44] X. Liu, L. Ma, X. Kong, and K. Y. Lee, "An efficient iterative learning predictive functional control for nonlinear batch processes," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4147–4160, 2020.
[45] M. Schwegel and A. Kugi, "A simple computationally efficient path ILC for industrial robotic manipulators," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2133–2139.
[46] Y. Chen, B. Chu, and C. T. Freeman, "Iterative learning control for robotic path following with trial-varying motion profiles," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4697–4706, 2022.
[47] B. Chu, A. Rauh, H. Aschemann, E. Rogers, and D. H. Owens, "Constrained iterative learning control for linear time-varying systems with experimental validation on a high-speed rack feeder," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1834–1846, 2021.
[48] R.-E. Precup, S. Preitl, J. K. Tar, M. L. Tomescu, M. Takács, P. Korondi, and P. Baranyi, "Fuzzy control system performance enhancement by iterative learning control," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3461–3475, 2008.
[49] A. Steinhauser and J. Swevers, "Iterative learning of feasible time-optimal trajectories for robot manipulators," in *The 20th World Congress of the International Federation of Automatic Control*, Toulouse, France, 2017, pp. 12 095–12 100.
[50] Y. Chen, B. Chu, and C. T. Freeman, "Point-to-point iterative learning control with optimal tracking time allocation," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1685–1698, 2017.
[51] T. Lippa and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.
[52] D. Verscheure, B. Demeulenaere, J. Swevers, J. de Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.
[53] J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.
[54] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in Matlab," in *In Proceedings of the CACSD Conference*, 2004, pp. 284–289.
[55] Y. Chen, B. Chu, and C. T. Freeman, "A coordinate descent approach to optimal tracking time allocation in point-to-point ILC," *Mechatronics*, vol. 59, pp. 25–34, 2019.
[56] J. D. Ratcliffe, "Iterative learning control implemented on a multi-axis system," Ph.D. dissertation, University of Southampton, Southampton, June 2005.
[57] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[58] S. J. Wright, "Coordinate descent algorithms," *Mathematical programming*, vol. 151, no. 1, pp. 3–34, 2015.

Yiyang Chen (Member, IEEE) received the M.Eng. degree from Imperial College London, London, U.K., in 2013, and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2017. After that, he worked as a Research Fellow in control systems (2017–2018) and in traffic signal control (2018–2020) at the University of Southampton. He joined the School of Mechanical and Electrical Engineering, Soochow University, in 2020, as an Associate Professor. He has published several papers in top control conferences and journals. His research interests include iterative learning control, optimization, artificial intelligence, image processing, and robotic systems.



Yiming Wang received the B.Eng. degree degree in the School of Mechanical and Electrical Engineering, Soochow University, in 2022. He is currently pursuing his MS degree in control engineering from Soochow University, Suzhou, China. His current research interests include iterative learning control and optimization.

Christopher T. Freeman received the B.Eng. degree in electromechanical engineering and the Ph.D. degree in applied control from the University of Southampton, Southampton, U.K., in 2000 and 2004, respectively, and the B.Sc. degree in mathematics from The Open University, Milton Keynes, U.K., in 2006. He is currently a Professor and Deputy Head of School within Electronics and Computer Science at the University of Southampton. He has authored 310 journal and conference papers primarily on the development, application, and assessment of intelligent controllers within both the biomedical engineering domain and for application to industrial systems.